

Bicubic Patch Surfaces for High-Speed Numerical Control Processing

Abstract: Parametric bicubic patch surfaces have been used for some time in manufacture and design. It is convenient to have such surfaces available as standard numerical control surfaces using the APT programming language. A major drawback is that they are costly to use for data processing of numerical control programs. If, however, nonparametric bicubics are used, computer time, and hence cost, can be reduced dramatically. This paper details a strategy and algorithms for this purpose. Experimental data suggest that computer costs are comparable to, or somewhat lower than, the costs for processing tabulated cylinder surfaces.

1. Introduction

The kinds of parametric bicubic patch surfaces [1-3] discussed in this paper have been used for some time in design and manufacture of complex surface configurations. Two approaches have been taken when these surfaces are involved in numerical control manufacture in conjunction with APT processing [4]. In one approach, cutter offset positions are calculated taking into account only the bicubic patch surface, and these points are sorted or ordered to produce a resultant cutter path which typically will machine the entire surface. In some cases additional logic is used to bound the final cutter path so that only a portion of the surface is machined. This approach is taken by the well known APTLOFT-FMILL package, a single-purpose program that computes only one surface and ignores the others in a given package of related surfaces.

In the second approach, the surface is used as one of two controlling surfaces which together determine the path of the cutter through space. This is the preferred technique, since the bicubic surface is a standard APT control surface and the full power and flexibility of the numerical control programming language may be used. A major drawback to this approach is that such surfaces are costly to use for processing numerical control data. The reason for this is that the basic problem faced in preparing the control tapes concerns the intersection of such a surface with a straight line. Moreover, it is not unusual to solve this problem 100000 to 1000000 times during the course of APT [4] processing. Elsewhere [5], devices have been developed for reducing the number of patches that need to be investigated.

If, however, nonparametric bicubic patch surfaces as in the F-mesh system of Dr. Inaba at Fujitsu Ltd. [6]

are used, the basic problem becomes completely manageable and very large reductions are obtained in computer storage and time, and hence in cost. There is, of course, a price to be paid for these reductions. The surface to be represented must be single-valued with respect to some plane. More precisely, it must have a finite bound for its maximum slope relative to some plane. This requirement excludes essentially multiple-valued surfaces, for example. In this system such surfaces would have to be represented by several single-valued segments whose boundaries projected on an associated plane are rectangular. In general these segments will overlap; hence for subsequent applications unique subsegments must be delimited by including bounding surfaces for each segment. Note that the total of the unique subsegments (except for common boundaries) must provide a complete covering of the total surface.

Since the surface intersection problem is the key to high-speed APT calculations, this paper is devoted to efficient solution of this problem and to testing the algorithms developed in a number of standard APT conditions.

As in the F-mesh system, the set of points used to define the nonparametric bicubic patch surface have the property that their projection on some plane defines a rectangular grid whose grid lines in general are not equally spaced. This raises a problem if the points provided do not have the requisite property. The solution to this problem requires interpolation and extrapolation from the given data to produce suitable points. Once an appropriate array of points is provided, the strategy used here to reduce computing time involves a sequence of successively more restrictive patch rejection techniques. For

those patches that remain after these tests, a sixth-order equation in one variable must be solved. Methods for dealing rapidly with this problem are also developed.

The first patch rejection technique comes from the obvious fact that if the straight line is projected onto the plane containing the rectangular grid, only those rectangles intersected by the projected line are projections of surface patches which may in fact be intersected by the line. The algorithm for this procedure is sufficiently obvious to require no further discussion. In sections 3 and 4 certain bounds are developed that provide rectangular parallelepipeds within which each patch is contained. One is also provided for the entire surface. In section 5 the problem of solving polynomial equations is discussed for the special circumstances arising in the problem at hand. In section 6 the techniques for using the bounds previously developed are exhibited.

In section 7 devices for bounding the number of zeros of the polynomial are developed. In section 8 an interval halving procedure is given for isolating zeros of the polynomial. Section 9 provides the results of extensive tests performed using this system. Comparisons of computer-time with the surface in APT are provided.

2. Notation

The set of bicubic patches is defined over a rectangular grid that is defined by the grid lines

$$\begin{aligned} x'_1, x'_2, \dots, x'_N, \\ y'_1, y'_2, \dots, y'_M. \end{aligned} \quad (1)$$

The general form of patch i, j , that is, the surface defined over the rectangular region

$$R'_{ij} [x'_i \leq x' \leq x'_{i+1}, y'_j \leq y' \leq y'_{j+1}], \quad (2)$$

is

$$z' = U(x') E_{ij} U^T(y'), \quad (3)$$

where $U(x')$ is a vector function $U(x) = (x^3, x^2, x, 1)$, U^T is its transpose, and E_{ij} is a 4×4 matrix of constants. With the transformation

$$\begin{aligned} x &= (x' - x'_i) / \delta_1, \delta_1 = x'_{i+1} - x'_i; \\ y &= (y' - y'_j) / \delta_2, \delta_2 = y'_{j+1} - y'_j; \\ z &= z', \end{aligned} \quad (4)$$

the equation has the form

$$z = U(x) M B_{ij} (U(y) M)^T \quad (5)$$

for x, y in $R[0 \leq x \leq 1, 0 \leq y \leq 1]$, where

$$M = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$B_{ij} = \begin{pmatrix} z_{0,0} & z_{0,1} & z_{y,0,0} & z_{y,0,1} \\ z_{1,0} & z_{1,1} & z_{y,1,0} & z_{y,1,1} \\ z_{x,0,0} & z_{x,0,1} & z_{x,y,0,0} & z_{x,y,0,1} \\ z_{x,1,0} & z_{x,1,1} & z_{x,y,1,0} & z_{x,y,1,1} \end{pmatrix}. \quad (6)$$

Here $z_{0,0} = z(0, 0)$, $z_{x,0,1} = \partial z(0, 1) / \partial x$, etc. Also $\partial x / \partial x = \delta_1 \partial z' / \partial x'$, $\partial^2 z / \partial x \partial y = \delta_1 \delta_2 \partial^2 z' / \partial x' \partial y'$, etc. Note that the components of B , also δ_1, δ_2 , are functions of i and j .

The line is defined by a point $p'_0 = (x'_0, y'_0, z'_0)$ and a vector $q'_0 = (u'_0, v'_0, w'_0)$. The equation of the line, then, is

$$p' \equiv (x', y', z') = p'_0 + t q'_0. \quad (7)$$

When dealing with the local coordinates x, y, z of a specific patch, the line equation must, of course, also be transformed to local coordinates. Its equation becomes

$$p \equiv (x, y, z) = p_0 + q_0 t, \quad (8)$$

where

$$\begin{aligned} p_0 &= (\alpha_1, \alpha_2, \alpha_3) \equiv ((x'_0 - x'_i) / \delta_1, (y'_0 - y'_j) / \delta_2, z'_0), \\ q_0 &= (\beta_1, \beta_2, \beta_3) \equiv (u'_0 / \delta_1, v'_0 / \delta_2, w'_0). \end{aligned} \quad (9)$$

3. Patch bounds and patch slope bounds

The i, j patch is already bounded by the four planes $x = x_i, x = x_{i+1}, y = y_j, y = y_{j+1}$. The present objective is to find a pair of parallel slant planes such that the parallelepiped formed by the six planes contains the surface segment in its interior; that is, to find $\alpha', \beta', \gamma'_2, \gamma'_1$ such that

$$\alpha' x' + \beta' y' + \gamma'_2 \leq z'(x', y') \leq \alpha' x' + \beta' y' + \gamma'_1 \quad (10)$$

for x', y' in R' , where the constants $\alpha', \beta', \gamma'_1, \gamma'_2$ are functions of i, j .

In the transformed coordinate system the desired inequalities are

$$ax + by + c_2 \leq z \leq ax + by + c_1 \quad (11)$$

over R_1 , where a, b, c_1, c_2 are functions of i, j .

Now consider the function

$$z^*(x, y) = z(x, y) - ax - by - c. \quad (12)$$

Clearly

$$z^*(x, y) = U(x) M B^* M^T U^T(y), \quad (13)$$

where

$$B^* = \begin{pmatrix} z_{0,0} - c & z_{0,1} - b - c & z_{y,0,0} - b & z_{y,0,1} - b \\ z_{1,0} - a - c & z_{1,1} - a - b - c & z_{y,1,0} - b & z_{y,1,1} - b \\ z_{x,0,0} - a & z_{x,0,1} - a & z_{x,y,0,0} & z_{x,y,0,1} \\ z_{x,1,0} - a & z_{x,1,1} - a & z_{x,y,1,0} & z_{x,y,1,1} \end{pmatrix}. \quad (14)$$

That is to say, having selected a, b, c by any means, the bounding plane problem for a given bicubic can be con-

verted to a simple bounding problem for a related bicubic. That is, two numbers c_1 and c_2 are now required such that

$$c_1 \geq z^*(x, y) \geq c_2 \text{ for } (x, y) \in R. \quad (15)$$

As to the choice of a, b, c with which to initiate the problem, it might be desirable to minimize the maximum directed derivative, since Taylor series expansions about various points will be used, and this would tend to make the first-order terms small. However, this is computationally infeasible, and may very well be not better than a simpler procedure, such as minimizing

$$\int_R \left[\left(\frac{\partial z^*}{\partial x} \right)^2 + \left(\frac{\partial z^*}{\partial y} \right)^2 \right].$$

Now from Eq. (12) the objective function becomes

$$\int_R \left[\left(\frac{\partial z}{\partial x} - a \right)^2 + \left(\frac{\partial z}{\partial y} - b \right)^2 \right]. \quad (16)$$

Thus it is a nonnegative quadratic function of a and b . It is evident that the minimum is zero if and only if the surface is a plane. In any case the minimum is attained at the (unique) values for a and b obtained by setting the respective derivatives of the objective function equal to zero. The result is

$$\begin{aligned} a &= \int_0^1 \int_0^1 \dot{u}(x) MBM^T u(y) = eBh^T \\ b &= \int_0^1 \int_0^1 u(x) MBM^T \dot{u}(y) = hBe^T \end{aligned} \quad (17)$$

where $\dot{u}$ is the derivative of u and

$$e = (-1, 1, 0, 0), \quad h = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{12}, -\frac{1}{12} \right). \quad (18)$$

For the subsequent development there is no loss of generality in setting $c = 0$.

Given a function $z(x, y)$ which is continuous together with all its derivatives (this is certainly true for bicubics), its extrema over a region R can be found by considering extrema on the boundary of R and the relative extrema inside R . It is not, in general, computationally feasible to find the relative extrema of a bicubic, in the amount of available computing time. Thus no attempt is made to find best bounds, but bounds as close as desired can be found [7, 8].

The maximum slope of the patch over $R'_{i,j}$ is

$$\sigma = \max \sqrt{(\partial z' / \partial x')^2 + (\partial z' / \partial y')^2}. \quad (19)$$

The transformation of the previous section ($R'_{i,j}$ to R), converts the problem to

$$\sigma = \max \sqrt{\left(\frac{\partial z}{\partial x} \right)^2 / \delta_1^2 + \left(\frac{\partial z}{\partial y} \right)^2 / \delta_2^2}. \quad (20)$$

The maximum is computed numerically as in [7], [8].

4. Total surface bounds

A bound for the maximum surface slope is simply the largest of the individual patch bounds.

To find parallel bounding planes compute the least square plane for the surface knots; that is, compute A, B, C such that

$$T = \sum_{i,j} (Ax_i + By_j + C - z_{i,j})^2 \quad (21)$$

is minimized. Then C_1 is chosen so that

$$Ax_i + By_j + C_1 \geq ax_i + by_j + c_1 \quad (22)$$

for all upper bounding planes at the knot x_i, y_j , for all knots. Similarly C_2 is chosen so that

$$Ax_i + By_j + C_2 \leq ax_i + by_j + c_2 \quad (23)$$

for all lower bounding planes at the knots.

5. Zeros of a polynomial

It will be observed in the following sections that the intersection problem reduces mathematically to solving an equation

$$f(x) = 0, \quad 0 \leq x \leq 1, \quad (24)$$

where x is a polynomial (of degree six or less in the present case). It is also observed that $f(x)$ measures the vertical distance between the bicubic and the line. Thus, in practice, a set of x 's is required such that each x is on $(0, 1)$ and

$$|f(x)| \leq \epsilon, \quad (25)$$

where ϵ is related to the accuracy of measurement of system data, that is, input points and line slope. Such x 's are called pseudozeros of $f(x)$.

Unless all x 's satisfying the inequality are included, which is clearly pointless, the necessarily finite set accepted is not uniquely defined. This is a matter of no moment for the problem at hand. The algorithm used is as follows. For $i = 0, 1, \dots, r-1$ let x_i be a value of $f_i(x)$ such that

$$|f_i(x_i)| \leq \delta \quad (26)$$

and define

$$f_{i+1}(x) = (f_i(x) - f_i(x_i)) / (x - x_i), \quad (27)$$

it being assumed that $f_0(x) \equiv f(x)$. For the index r it is true that either $|f(x)| > \delta$ or that $f(x)$ does not change sign in the closed interval. Clearly, $r \leq n$ the degree of the polynomial f .

Let

$$w_0(x) \equiv 1,$$

$$w_i(x) \equiv (x - x_0)(x - x_1) \cdots (x - x_{i-1}). \quad (28)$$

It follows that

$$f_0(x) = P(x) + \sum_{i=0}^{r-1} w_i(x) f_i(x_i), \quad (29)$$

where

$$P(x) = w_r(x) f_r(x), \quad (30)$$

and $P(x) \equiv f_0(x)$ in the event that $|f_0(x)| > \sigma$ for [9] $x \in [0, 1]$.

Now $|w_i(x)| \leq 1$ for all i , since x and x_i are all in $[0, 1]$. Hence

$$|f(x) - P(x)| \leq r\delta \leq \epsilon \quad (31)$$

when δ is chosen less than ϵ/n . Hence $x_0, x_1, \dots, x_{r-1}$, the zeros of $P(x)$, are pseudozeros of $f(x)$.

Graphically the initial problem involves a graph of $f(x)$ on $[0, 1]$, and a band of width 2ϵ centered on the x axis. Whenever the graph enters the band it may or may not emerge, and if it does it will emerge either on the side of entry, or on the opposite side. The first two cases may or may not be represented by a pseudozero. If ϵ has been reasonably chosen it does not matter. The last case should certainly be recognized by $P(x)$ and is, as a matter of fact. For if $f(x') \geq \epsilon$, $f(x'') \leq -\epsilon$ the inequality (31) requires that $P(x') \geq 0$, $P(x'') \leq 0$, so that at least one of the x_j 's is in the closed interval defined by x' , x'' . This is the only unequivocal statement that can be made about the relative locations of the mathematical zeros of $f(x)$ and of $P(x)$. As to the number of zeros, $P(x)$ may have either more zeros than $f(x)$ or less. The only certainty about the number of zeros of $P(x)$ is that it does not exceed n . Neither the number of pseudozeros of $f(x)$ nor their magnitude can be determined in advance.

6. Initial procedures for locating intersections

Given a line L , and

$$p' = p'_0 + tq'_0, \quad (32)$$

it may be possible to determine, using the bounding planes for the entire bicubic surface, and the x' , y' boundaries of that surface, that the line cannot intersect the surface.

If this is not the case, then it will be necessary to deal successively with the set E of patches selected as candidates by the projected-line technique mentioned in the Introduction. There are three situations involving slopes that may simplify the problem considerably. Let

$$\sigma = |w'_0| / \sqrt{(u'_0)^2 + (v'_0)^2} \quad (33)$$

be the minimum slope of the line, σ^* be the bound for the maximum slope of the entire surface, σ_E be the maximum slope of the set E of patches. If either

$$(w'_0)^2 > [(u'_0)^2 + (v'_0)^2] (\sigma^*)^2 \quad (34)$$

or

$$(w'_0)^2 > [(u'_0)^2 + (v'_0)^2] \sigma_E^2, \quad (35)$$

then there is at most one intersection of the line with the entire surface. The difference between the two cases lies in the way they are to be programmed, since σ^* is available at once, and σ_E is not. In both cases, once a solution is found, the problem is solved. In the first case the problem is terminated when and if a solution is found. In the second case, it is necessary to examine all the patches of E to verify the stated condition, but only necessary to look for a solution until one is found, if it exists.

If neither condition holds, it is still true that in any patch for which

$$(w'_0)^2 > [(u'_0)^2 + (v'_0)^2] \sigma_p^2, \quad (36)$$

where σ_p is the patch slope bound, at most one solution exists.

If a patch is a member of the set E , it remains a candidate patch if and only if the line L intersects two sides of the parallelepiped containing the patch. If it does not, the patch is no longer a candidate. If it does, then two values t_1 and t_2 of the line parameter t are determined such that any intersection of the line and surface corresponds to a value of t between t_1 and t_2 .

Now, in terms of the local coordinate system, the line equations are

$$\begin{aligned} x &= \alpha_1 + \beta_1 t, \\ y &= \alpha_2 + \beta_2 t, \\ z &= \alpha_3 + \beta_3 t, \end{aligned} \quad (37)$$

and the patch equation is

$$z = U(x) S U^T(y), \quad S = M B_{i,j} M^T. \quad (38)$$

For a given value of t ,

$$\begin{aligned} z_{\text{line}} &= z_0 + w_0 t, \\ z_{\text{patch}} &= U(x_0 + u_0 t) S U^T(y_0 + v_0 t), \\ z_{\text{patch}} - z_{\text{line}} &= \phi(t) \equiv U(x_0 + u_0 t) S U^T(y_0 + v_0 t) \\ &\quad - z_0 - w_0 t. \end{aligned} \quad (39)$$

Thus $\phi(t)$ is the vertical distance between the patch and the line, at the values of x_0 , y_0 specified by t . The distance from the line normal to the surface is less than $|\phi(t)|$, and hence less than ϵ whenever $|\phi(t)|$ is.

If it is guaranteed by slope conditions that at most one zero exists, then it is necessary only to compute $\phi(t_1)$ and $\phi(t_2)$. Then either $\text{sign } \phi(t_1) = \text{sign } \phi(t_2)$, and no zero exists, or $\text{sign } \phi(t_1) \neq \text{sign } \phi(t_2)$ and exactly one zero exists.

In this last case the explicit polynomial $\phi(t)$, of order six or less, is computed, and a pseudozero is found, per-

haps by a modified regula falsi [7, 8]. Note that in this case a pseudozero is allowed to exist if and only if a zero of $\phi(t)$ is known to exist.

The explicit polynomial is

$$\phi(t) = U(t)B_1A_1SA_2B_2U^T(t) - \beta_3t - \alpha_3, \quad (40)$$

where

$$A_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3\alpha_i & 1 & 0 & 0 \\ 3\alpha_i^2 & 2\alpha_i & 1 & 0 \\ \alpha_i^3 & \alpha_i^2 & \alpha_i & 0 \end{pmatrix}, B_i = \begin{pmatrix} \beta_i^3 & 0 & 0 & 0 \\ 0 & \beta_i^2 & 0 & 0 \\ 0 & 0 & \beta_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (41)$$

It may be useful to note that if either β_1 or $\beta_2 = 0$, then $\phi(t)$ reduces to a cubic.

As to computing costs, computing $\phi(t)$ without computing its coefficients in general requires 21 multiplications and 17 additions per t value. The computation of the explicit polynomial (assuming sixth order) requires 85 multiplications and 63 additions.

Thus, if it is not true a priori that the number of zeros is bounded by one, the explicit polynomial should be computed immediately, followed by the transformations of the next section.

7. Polynomial transformations

The present objective is to obtain a bound for the number of zeros in the interval (t_1, t_2) . For this purpose it is convenient to make a linear transformation of variables that carry (t_1, t_2) into $(0, 1)$. That transformation is

$$t = x(t_2 - t_1) + t_1. \quad (42)$$

It is convenient to perform the calculation in two steps, thus

$$\theta(s) = \phi(s + t_1), s \in (0, t_2 - t_1), \quad (43)$$

at a computing cost of $n(n+1)/2$ multiplications and a like number of additions, where n is the degree of the polynomial, followed by

$$f(x) = \theta[x(t_2 - t_1)], x \in (0, 1), \quad (44)$$

for which the cost is $2n-1$ multiplications.

At this point all the 0, 1 pseudozeros of $f(x)$ are removed. The result is a polynomial $f(x)$ of lower degree, for which $|f(0)| > \epsilon$, $|f(1)| > \epsilon$. If $\text{sign } f(0) \neq \text{sign } f(1)$ a pseudozero exists between 0 and 1. Compute it and reduce $f(x)$ again.

In any event, there ultimately results an $f(x)$ having the properties

$$|f(0)| > \epsilon, |f(1)| > \epsilon, \text{sign } f(0) = \text{sign } f(1), \quad (45)$$

and there are evidently an even number of zeros of this $f(x)$.

Now let $z' = z/(1-z)$ and $g(z') = (1+z')^n f[z'/(1+z')]$, at a cost of $n(n+1)/2$ additions.

This is a transformation which, in the complex plane, explodes the circle centered at $(1/2, 0)$, radius $1/2$, into the right half plane, and carries reals into reals. The effect is that the number of positive real zeros of $g(z')$ is precisely the same as the number of real zeros of $f(x)$ for $x \in [0, 1]$. However, the number of positive real zeros of $g(z')$ is bounded by the number of changes of sign among consecutive coefficients of $g(z')$, and differs from that number by an even number.

The principal reason for the use of this procedure is to be found in the results of the experiment reported here. The experiment consisted of forming sixth-order polynomials with random coefficients uniformly distributed over $(-1, 1)$, discarding those polynomials which had zeros on the interval $[0, 1]$, and carrying the rest through the above procedure. The results for 5000 acceptable polynomials was 4774 cases with no sign changes, 225 cases with two sign changes, and one case with four sign changes; that is, if the polynomial had no zeros on the interval $[0, 1]$ the experimental probability that there would be no sign change in $g(z')$ was 0.945. For polynomials of lower degree the experimental probability was somewhat higher.

In view of the hypothesis concerning the relative rarity of multiple intersections, it seems reasonable to assume that this test will, in nearly 19 cases out of 20, exclude the patch from further consideration. It has the further advantage that if there are, in fact, multiple intersections, the patch will be retained for further analysis.

The effects of these calculations on error in the various derived numbers seems hardly worth consideration, since roundoff errors in the computer are completely trivial when compared to measurement errors inherent in the physical data, particularly in view of the small number of operations involved.

8. Interval halving

The preceding calculations result either in eliminating the patch, or in its complete solution, or in the present problem. That is, given a polynomial $f(x)$ of degree six or less, of which it is known that

$$\begin{aligned} \text{sign } f(0) &= \text{sign } f(1), \\ |f(0)| > \epsilon, |f(1)| > \epsilon, \end{aligned} \quad (46)$$

and that an even number of zeros on $(0, 1)$ is possible, isolate those zeros if they exist. It will simplify matters to assume that $f(0) > \epsilon$, $f(1) > \epsilon$ which requires, at most, multiplying $f(x)$ by -1 .

The following procedure always locates a point at which $f(x) \leq 0$, if one exists. If no such point exists, then $f(x) > 0$ throughout the interval, and the procedure may or may not locate a point such that $f(x) \leq \epsilon$. In outline, the procedure performs as follows. It begins with the interval $(0, 1)$ and proceeds to form a list of intervals.

If (x_1, x_2) is an interval belonging to the list, its analysis consists of the following steps.

1. A value of x is located which isolates two zeros of $f(x)$, and the procedure is terminated.
2. It is determined that $f(x) > \epsilon$ throughout the interval so that the interval is deleted from the list, and the half intervals $[x_1, (x_1 + x_2)/2]$, $[(x_1 + x_2)/2, x_2]$ need not be examined.
3. Neither situation occurs and no conclusion is possible for the entire interval. It is removed from the list and the half intervals examined. If case 1 occurs, the procedure is terminated. If case 2 occurs, the half interval involved is not added to the list. If case 3 occurs, the half interval involved is added to the list.

Thus the result of the analysis is that (x_1, x_2) is replaced by zero, one, or two half intervals.

The list is initiated with the interval $(0, 1)$. The procedure terminates when a value of x is found which isolates two zeros, or when the list is empty, or when it can be asserted that $f(x) > 0$ in all the intervals remaining in the list. It is this last assertion that assures the finiteness of the algorithm.

The basis of the procedure is the equation [10]

$$f(x) = P(x) + f'''(\xi)(x - x_0)(x - x_1)(x - x_2)/6, \quad (46)$$

where $x_0 < x_1 < x_2$, $x_0 < \xi < x_2$, $P(x)$ is a quadratic polynomial passing through the points $(x_i, f(x_i))$ $i = 0, 1, 2$, and f''' is the third derivative of f . With

$$x_1 = (x_0 + x_2)/2, \Delta = (x_2 - x_0)/2, u = x_1 - x,$$

$$y_i = f(x_i)$$

$$A = y_2 - 2y_1 + y_0, B = y_0 - y_2, C = y,$$

Equation (46) becomes

$$f(x) = P(x) + f'''(\xi)(u - \Delta)u(u + \Delta)/6$$

$$P(x) = (A/2\Delta^2)u^2 - (B/2\Delta)u + C. \quad (47)$$

Let

$$M = \max|f'''(x)|, x \in (0, 1)$$

and note that for $|u| \leq \Delta$

$$1/6 |(u - \Delta)u(u + \Delta)| < \alpha = 0.06415004,$$

whence

$$P_{00}(x) = P(x) - \alpha M \Delta^3 \leq f(x) \leq P(x) + \alpha M \Delta^3$$

$$= P_{01}(x).$$

If, for $|u| \leq \Delta$, $\min P_{00}(x) > \epsilon$ the interval may be discarded. If, for $|u| \leq \Delta$, $\min P_{01}(x) \leq \epsilon$ then a pair of pseudozeros has been located and the procedure terminates. If neither condition is true, the half intervals will be analyzed. Before undertaking these tests, however,

test $y_1 < \epsilon$ which, if true, locates two pseudozeros and terminates the process. Otherwise note that if $A \leq 0$, or $A > 0$ and $|B| \geq 2A$ the minima occur at either x_0 or x_2 , where $y_0 > \epsilon$, $y_2 > \epsilon$. Finally, if the minima require investigation

$$\min P_{00} = C - (B^2/8A) - \alpha M \Delta^3,$$

$$\min P_{01} = C - (B^2/8A) + \alpha M \Delta^3.$$

If the test of the entire interval is inconclusive, note that

$$0 \leq (u - \Delta)u(u + \Delta) \leq -2\Delta u(u + \Delta), \quad u \in (-\Delta, 0)$$

$$0 \geq (u - \Delta)u(u + \Delta) \geq 2\Delta u(u - \Delta), \quad u \in (0, \Delta).$$

For the interval $(-\Delta, 0)$ let

$$P_{10}(x) = (A_{10}/2\Delta^2)u^2 - (B_{10}/2\Delta)u + C,$$

$$P_{11}(x) = (A_{11}/2\Delta^2)u^2 - (B_{11}/2\Delta)u + C,$$

where

$$A_{10} = A + a, B_{10} = B - a, A_{11} = A - a, B_{11} = B + a,$$

$$a = 2M\Delta^3/3.$$

Now

$$P_{10}(x) \leq f(x) \leq P_{11}(x), \quad x_0 \leq x \leq x_1,$$

$$P_{10}(x) = f(x) = P_{11}(x), \quad x = x_0 \text{ or } x_1.$$

For this case

1. Either $-2A_{10} < B_{10} < 0$, or the interval can be discarded, or
2. If $-2A_{10} < B_{10} < 0$, $P_{10}(x)$ has a minimum at $u = B_{10}\Delta/2A_{10}$ of magnitude $C - (B_{10}^2/8A_{10})$.

If this number is greater than ϵ , the interval can be discarded. Otherwise,

3. either $-2A_{11} < B_{11} < 0$, and no conclusion can be reached concerning this interval,
4. or the minimum occurs at $(B_{11}\Delta/2A_{11})$ and has magnitude $C - (B_{11}^2/8A_{11})$. If this number is not greater than ϵ , a pair of zeros or pseudozeros has been isolated in the present interval. Otherwise no conclusion is possible, in which event the interval is appended to the list. If $C - (B_{10}^2/8A_{10}) > 0$ this fact is noted and also saved.

For the interval $0 \leq u \leq \Delta$, with

$$P_{20}(x) = (A_{20}/2\Delta^2)u^2 - (B_{20}/2\Delta)u + C,$$

$$P_{21}(x) = (A_{21}/2\Delta^2)u^2 - (B_{21}/2\Delta)u + C,$$

where

$$A_{20} = A_{10}, B_{20} = B_{11}, A_{21} = A_{11}, B_{21} = B_{10}$$

the conditions become

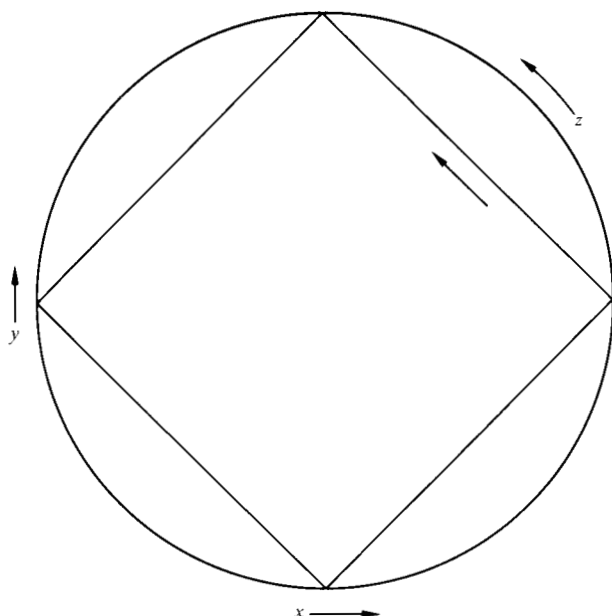


Figure 1 Diamond-circle pattern produced by basic part program used for the test cases.

1. $0 < B_{20} < 2A_{20}$ not true, discard the interval,
2. otherwise test $C - B_{20}/8A_{20} > \epsilon$. If true, discard the interval. If not true, then
3. test $0 < B_{21} < 2A_{21}$. If not true, no conclusion can be reached regarding this interval. If this is true, then
4. compute $C - (B_{21}^2/8A_{21})$. If the number is not greater than ϵ then a pair of zeros or pseudozeros has been isolated. If greater than ϵ , no conclusion is possible for this interval, and it is added to the list, together with an appropriate indicator if $C - (B_{20}^2/8A_{20})$ is positive.

It is necessary to maintain two counts, one being the number of intervals in the list, the other being the number of intervals in which $f(x)$ is entirely positive. If the first is zero, or is equal to the second number, the procedure is terminated.

Experience to date suggests that the number of intervals tested per problem averages about three. For example, for the function $f(x) = (x - 0.5)^6$ the zero 0.5 is found immediately. For $f(x) = (x - 1/\sqrt{2})^6 + 10^{-7}$, $\epsilon = 10^{-9}$, about 150 iterations were required to discard the function. For $f(x) = (x - 1/\sqrt{2})^6 + 10^{-9}$ with $\epsilon = 10^{-7}$, the first pseudozero was found after about the same number of iterations. These would appear to be fairly representative of worst cases.

9. Experimental results

A principal objective of this work was to reduce computer processing times, and hence costs, when dealing with bicubic patch surfaces. In order to obtain a meaning-

ful measure of the performance, it was decided to compare the APT processing times (for Section 2, the arithmetic element) of rectangular bicubic patch surfaces with standard APT surfaces. The tabulated cylinder (a cylinder with a piecewise continuous planar spline curve generator and a straight-line generatrix) was chosen for comparison purposes, since this is one of the more complex types of APT surface but is none the less sufficiently economical that it is widely used. This surface type is abbreviated TC in the following discussion.

Code modifications and additions to implement the bicubic patch surface algorithms described in the preceding sections were made to the APT system as implemented under CMS [11], and a series of tests was run.

The same basic APT part program was used for all test cases. This program required the cutter to be driven around a series of planes and a right circular cylinder, so that the xy projection of the resultant cutter path produced the 'diamond-circle' pattern shown in Fig. 1. The z coordinates of the resultant cutter path were controlled by the part surface that was varied between test series.

Each of the first four test series used two part surfaces that were very similar in overall shape—one which was defined as a TC surface and the other which was defined as a bicubic patch surface. The number of TC intervals (the number of intervals on the spline defining the TC) and the number of patches on the bicubic patch surface were the same within each test series and varied from series to series. The part surfaces used in the tests are described for each test.

A range of five different APT programming conditions was used in the tests. These differing conditions varied the complexity of the APT calculations required to compute the cutter path. The conditions used were:

APT programming condition

- | | |
|------------------------------------|--|
| A. (TLONPS, 3 axis) | Simplest case; cutter is effectively a single point which is to be on the part surface. Cutter axis is fixed. |
| B. (TLOFPS, Ball Cutter)
3 axis | Somewhat more complex; cutter end is a hemisphere which must be positioned tangent to the part surface. Cutter axis is fixed. |
| C. (TLOFPS, Flat Cutter)
3 axis | More complex; cutter is a section of a right circular cylinder which must be positioned tangent to the part surface. Cutter axis is fixed. |

- D. (TLOFPS, Ball Cutter) MULTAX Still more complex; cutter end is a hemisphere but the cutter axis must be lined up parallel to the surface normal at the point of tangency between the cutter and the surface.
- E. (TLOFPS, Flat Cutter) MULTAX Most complex, cutter is a section of a right circular cylinder which must be positioned tangent to the part surface with the tool axis oriented normal to the surface at the point of tangency.

Test series 1

A section of the parabolic cylinder $y = z^2$ (see Fig. 2) approximated by a four-interval TC and by a four-patch bicubic surface.

Test series 2

The same section of the parabolic cylinder $y = z^2$ was approximated by a 49-interval TC and by a 49-patch bicubic surface.

Test series 3

The same section of the parabolic cylinder $y = z^2$ was approximated by an 81-interval TC and by an 81-patch bicubic surface.

Test series 4

Two cycles of a cylindrical surface with a sine wave generator (see Fig. 3) approximated by a 60-interval TC and by a 60-patch bicubic surface.

Test series 5

The sine wave surface of the preceding series was distorted so it was no longer cylindrical, and approximated by a 40-patch bicubic surface (see Fig. 4). The purpose of this test was to determine whether the fact that the bicubic patches in the previous tests were all ruled surfaces was significantly affecting the timing. Processing time, as compared with the patch surface of series 4, was not substantially different.

The execution times for the tests are given in Table 1, which summarizes APT execution times.

With one exception, the bicubic surface was always processed in less time than the TC with the same number of intervals. In that one case (1 - D) the probable explanation for the difference is that minor variations in numerical values can cause significantly different

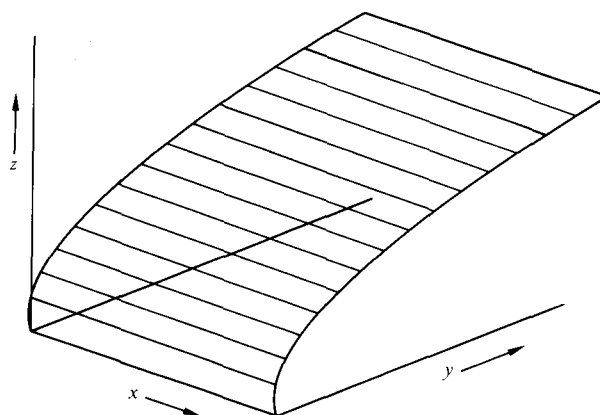


Figure 2 Section of a paraboloid cylinder approximated by a four-interval TC program and by a four-patch bicubic surface.

branching in the highly iterative procedures used by the arithmetic element of APT. This phenomenon is not unusual.

Increasing the complexity of the APT programming condition, again with the one exception, increased both the TC and bicubic surface time, with their ratio remaining constant for a given number of patches or intervals. The overall ratio of time intervals changed in favor of the BICUBIC surface (see Fig. 5) as the number of patches or intervals was increased.

Table 1 Execution times in tests comparing APT processing of rectangular cubic patch surfaces with the APT TC standard surface.

Test series	Programming condition	Time for TC part surface (seconds)	Time for bicubic part surface (seconds)	Ratio bicubic/TC
1	A	15.00	13.93	0.92
	B	19.80	19.02	0.96
	C	20.22	19.40	0.95
	D	19.82	32.18	1.62
	E	23.26	22.95	0.98
2	A	25.05	18.10	0.72
	B	33.74	24.53	0.72
	C	34.15	24.90	0.73
	D	36.66	25.50	0.69
	E	42.06	29.43	0.70
3	E	55.31	31.51	0.56
4	A	34.42	26.56	0.77
5	A	not applicable	30.92	not applicable

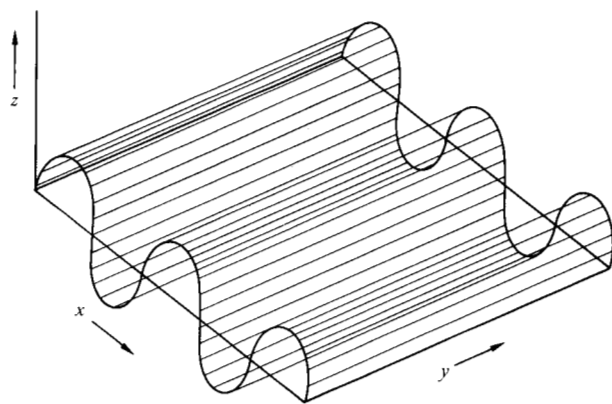


Figure 3 Two cycles of a cylindrical surface with a sine-wave generator approximated by a 60-interval TC program and by a 60-patch bicubic surface.

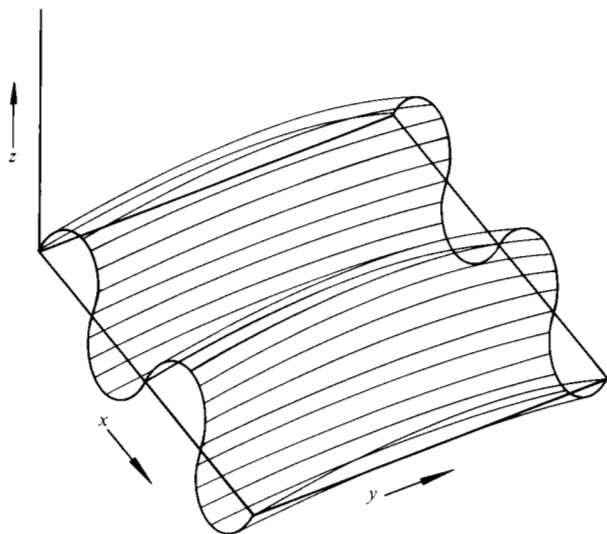


Figure 4 Sine-wave surface approximated by a 40-patch bicubic surface.

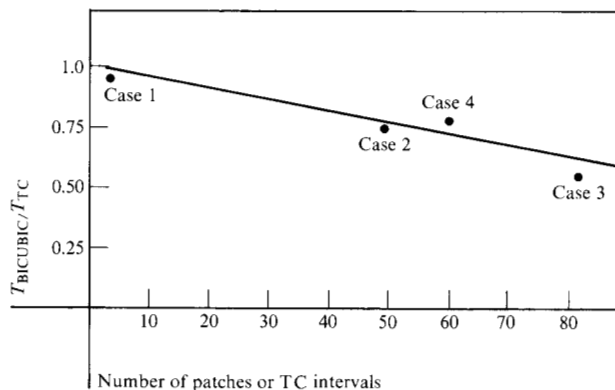


Figure 5 Progressive change in the ratio of $T_{BICUBIC}/T_{TC}$ processing-time interval as complexity of programming increases.

10. Summary

The necessity to deal with complex, doubly curved surfaces presents many problems to the overall design, engineering, and manufacturing cycle. One such problem is that when these surfaces are implemented in APT, large amounts of computer time are frequently needed to prepare numerical control data. The object of the investigation reported here was to determine the relative economic feasibility of using a nonparametric patch surface representation. This representation makes it possible to reduce the computer time required to do the fundamental calculation required for APT processing, that of intersecting a line with a surface.

This time reduction is possible because:

- A linearization technique eliminates many of the patches on a surface as candidates for intersection with a given line.
- A parallelepiped surrounding each of the remaining patches further reduces possible candidates and supplies bounds for the independent variable of the sixth-degree equation determining intersection with the line.
- For any remaining patches, an upper bound for the number of solutions is to be computed. Since bounds greater than one are expected to be rare, computing time is again relatively small.

To evaluate the relative economics, APT processing times for a nonparametric bicubic patch were compared with processing times for TC surfaces over a range of APT programming conditions. It was surprising to discover that the bicubic surface was less costly than TC in all but one instance, since the opposite had been expected. From the point of view of computing time, such bicubic surfaces are at least as economical as TC surfaces.

Obviously the surfaces investigated here are more restrictive than parametric bicubic patch surfaces. In particular, multiple-valued surfaces have to be segmented and represented in several segments, each of which is single-valued with respect to some plane. Such representations present problems that are far from trivial. These problems must be dealt with before this special representation can be considered for production usage with surfaces that require such treatment.

References and notes

1. J. Ferguson, "Multivariable Curve Interpolation," *J. ACM* 11, 221 (1964).
2. S. A. Coons, "Surfaces for Computer-Aided Design of Space Forms," *Project MAC*, Massachusetts Institute of Technology, Cambridge, MA, June 1967.
3. B. Dimsdale, "On Spline End Conditions," *Technical Report G320-2656*, IBM Scientific Center, Los Angeles, California 1972, pp. 1-10.
4. *System/360 APT Numerical Control*, Report GH20-0309-4, IBM Corporation, White Plains, New York 1970.

5. B. Dimsdale, "Bicubic Patch Bounds," *Technical Report G320-2665*, IBM Scientific Center, Los Angeles, California 1974.
6. P. Bézier, *Numerical Control Mathematics and Applications*, John Wiley and Sons, New York, 1970, pp. 152-161.
7. B. Dimsdale, "Straight Line Intersections with a Bicubic Patch Surface," *Technical Report G320-2657*, IBM Scientific Center, Los Angeles, California 1973.
8. R. P. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1973.
9. $P(x) = f_0(x)$ is a possibility (but not a certainty) if $f_0(x) \neq 0$.
10. A. S. Householder, *Principles of Numerical Analysis*, McGraw-Hill Book Company, Inc., New York, 1953, p. 196.
11. *APT Under CMS*, IBM Field Developed Program 5798-AGD, IBM Corporation, White Plains, New York 1972.

Received May 13, 1975

The authors are located at the IBM Los Angeles Scientific Center, 1930 Century Park West, Los Angeles, California 90067.